1-1-2002

A 2-secure code with efficient tracing algorithm

Joseph Tonien
University of Wollongong, dong@uow.edu.au

Rei Safavi-Naini
University of Calgary, rei@uow.edu.au

Yejing Wang
University of Wollongong

A 2-secure code with efficient tracing algorithm

A 2-Secure Code with Efficient Tracing Algorithm

Vu Dong Tô, Reihaneh Safavi-Naini, and Yejing Wang

School of Information Technology and Computer Science
University of Wollongong, Wollongong 2522, Australia
{dong,rei,yejing}@uow.edu.au

Abstract. Collusion secure fingerprinting is used to protect against illegal redistribution of digital documents. Fingerprints are embedded in documents to identify different copies. A group of colluders having access to multiple copies with different fingerprints may construct a pirate object with a fingerprint that cannot be traced. We consider c-secure codes with ε error that allow one of the c possible colluders to be traced and the chance of incorrect tracing to be at most ε. We consider a two layer construction consisting of an inner code and an outer structure and give new constructions for each. Important properties of our new inner code is that innocent users will never be accused and the code can be constructed for any number of codewords. This is particularly important as the number of codewords is the alphabet size of the outer structure. We will show that for the outer structure a c-traceability code, or a perfect hash family can be used and obtain the parameters of the combined code in terms of the parameters of the inner code and those of the outer structure. We apply these constructions to our new inner code and give parameters of the resulting c-secure codes.

Keywords: fingerprinting codes, frameproof codes, secure codes, secure frameproof codes, traceability codes.

1 Introduction

Fingerprinting is used to distinguish different copies of the same document or software. A fingerprint is a q-ary mark sequence that is embedded in the object in an imperceptible and robust (hard to remove) way. Collusion secure fingerprinting [3] aims at tracing pirate objects constructed by a collusion of users who have access to multiple copies of the same object, each with a different fingerprint.

To construct a pirate object, colluders compare their objects to find the places where their marks are different, and construct a pirate object by using one of their marks in each detected position. *Totally c-secure* codes allow one of the colluders to be traced if the size of the collusion is at most c. Boneh et al showed that totally c-secure codes do not exist for $c \geq 2$ and introduced c-secure codes with ε-error in which a member of collusion will be found with probability of at

least $1 - \varepsilon$. The ε-error refers to the error of the tracing algorithm. The error could be due to the failure of the algorithm to trace some of the pirate objects, or to output an innocent user in some cases. The latter case is undesirable for realistic scenarios and must be avoided.

Important parameters of c-secure codes are the length and the number of codewords. Good codes have shorter length and higher number of codewords.

The main construction of c-secure codes is due to Boneh et al. [3, 4] and consists of an outer code which is an error-correcting code, and an inner code. Other constructions retain this structure but give different construction for the inner code.

In this paper we present a number of new results on 2-secure codes. The main construction that we consider is a two level construction that consists of an inner code and an outer structure. The outer structure can be an error-correcting code, or a perfect hash family. The set of codewords of the inner code form the alphabet set of the outer code and so we require inner codes to be constructible for a wide range of alphabet sizes. In particular to compare two inner codes we will fix the size of the alphabet.

Firstly, we construct a new 2-secure inner code of length n^2 with n codewords and give an upper bound on ε which shows that the probability of error decreases exponentially with n. We give an efficient tracing algorithm for this code and show that the tracing algorithm never accuses an innocent user. That is either tracing algorithm fails and does not output, or else it outputs a colluder. An interesting property of the code is that for the same error probability it has shorter length when compared with the inner code in [3, 4], or [9] with the same number of codewords. Although the inner code in [3, 4] is only for c-secure codes with $c \geq 3$ but since a c-secure code is also a c'-secure for $c' < c$ we will compare our code with an instance of the code with the same number of codewords.

Then we consider possible outer structures. First, we show that using a 2-TA code as the outer structure combined with a 2-secure code with ε error results in a 2-secure code with ε' error and give the value of ε'. 2-TA codes can be obtained from error-correcting codes whose minimum distance satisfy a lower bound. We will show that equi-distance codes with odd minimum distance are 2-TA codes and can always be used for the outer code.

Next we show that perfect hash families (PHF) can be used as the outer structure to construct a c-secure code with more codewords from a smaller c-secure code. We will obtain probability of failure of tracing as a function of ε and s, the number of functions in the perfect hash family.

The tracing algorithm in the case of error-correcting codes consists of two stages: first using the decoding algorithm of the outer code followed by the tracing algorithm of the inner code. In the case of PHF as outer code, tracing consist of finding a function in the family that satisfies certain property followed by the tracing of the inner code. Efficiency of the former stage of tracing depends on the structure of PHF.

We will use both outer structures with our proposed inner code and obtain the parameters of the resulting codes. The final code in all cases will have the property that only colluders will be captured.

1.1 Related Works

Secure fingerprinting codes have been defined with a range of security properties.

Frameproof codes

Frameproof codes are introduced in [3], and constructed in [3, 4, 12, 13]. A c-frameproof code provides a property that any up to c colluders cannot create the fingerprint of an innocent user. Constructions of frameproof codes are given by [3, 12, 10].

Secure frameproof codes

A weak notion of secure codes is secure frameproof codes. A c-secure frameproof code is defined and constructed by Stinson et al in [11] requires that two disjoint collusions not be able to create the same pirate word. c-Secure frameproof code do not provide tracing algorithm and only require the structure of the code to support unambiguous tracing.

Traceability codes

Traceability codes are introduced by Staddon et al in [10]. A c-TA code provide frameproofness and traceability property. That is a group of up to c colluders cannot frame another user and any pirate word that they construct is closet to the codeword of one of the colluders and so a colluder can always be found by finding the codeword with minimum Hamming distance to the pirate word. c-TA codes can be constructed from error-correcting codes. For these codes tracing algorithm is the same as decoding algorithm of the error-correcting code. This is particularly useful for codes that have efficient decoding algorithm.

Traitor tracing schemes

Traitor tracing schemes are introduced in the context of broadcast encryption systems [6] and data fingerprinting [3]. In a broadcast encryption system, the sender broadcasts an encrypted message through a broadcast channel such that only members of an authorised group of receivers can decrypt the message. To do so, each receiver has a decoder with a unique key set. A group of colluders may use their key information to construct a pirate decoder that can decrypt the broadcast. Traitor tracing schemes allow one of the colluders to be identified when a pirate decoder is found. Known constructions of traitor tracing systems use combinatorial designs [12, 13], and error-correcting codes [12]. Tracing traitors in public key encryption systems are proposed in [2]. It is shown [7] that tracing is impossible when the number of traitors exceeds a certain number.

The rest of the paper is organised as follows. In Section 2 we recall the definitions and review the known results that will be used throughout the paper. In Section 3 we define a new inner code, provide an efficient tracing algorithm and show the properties of the code. We construct 2-secure codes by combining our new inner code with error-correcting codes in Section 4, and with perfect hash families in Section 5. Finally, we compare our constructions with existing ones and conclude the paper in Section 6.

2 Preliminaries

Let Γ be a q-ary code of length ℓ and size n. We have $\Gamma \subseteq Q^\ell$, where Q is a set of alphabets, $|Q| = q$, and $|\Gamma| = n$. An element of Γ, called a codeword, can be written as $w = (w_1, w_2, \ldots, w_\ell)$, where $w_i \in Q$. Elements of Q^ℓ in general are called words.

Let $C = \{w^{(1)}, w^{(2)}, \ldots, w^{(c)}\} \subseteq \Gamma$. A position i is called an *undetectable position* for C if $w_i^{(1)} = w_i^{(2)} = \ldots = w_i^{(c)}$; otherwise, it is called a *detectable position*. We denote the set of all undetectable and detectable positions for C as $U(C)$ and $D(C)$. Define the *descendant set* of C as

$$Desc(C) = \{w \in Q^\ell : w_i \in \{w_i^{(1)}, w_i^{(2)}, \ldots, w_i^{(c)}\},\ \forall i\}.$$

$Desc(C)$ is the set of all words that can be constructed by the coalition C. An element w of $Desc(C)$ is called a *descendant* of C and elements of C are called *parents* of w.

We use the following Marking Assumption and Embedding Assumption which were first introduced in [3].

Marking Assumption: A collusion is only capable of modifying detectable positions.

Embedding Assumption: A user has no knowledge of which mark in the object encodes which bit in the code.

Colluders can modify the symbols at detectable positions and can replace them with any symbol in the alphabet or replace it with an unrecognizable symbol, denoted by '?', that is not in the alphabet Q. We call $Q' = Q \cup \{?\}$ as the *extended alphabet*. And we define the *feasible set* $F(C)$ of C as

$$F(C) = \{w \in {Q'}^\ell : w_i = w_i^{(j)} \quad \forall i \in U(C), w^{(j)} \in C\}$$

If Γ is a binary code then $Desc(C)$ contains precisely all the elements of $F(C)$ that do not contain marks '?'. If an element of $F(C)$ contains '?' in some positions, if we substitute these marks '?' by any symbol of Q, then the resulting word must belong to $Desc(C)$. This is only true if the code Γ is binary.

Lemma 1. *Let Γ be a binary code, and $C \subseteq \Gamma$. For an element of $F(C)$, substituting all the marks '?' by 0 or 1 arbitrarily will result in an element of $Desc(C)$.*

Frameproof codes introduced in [3], ensure that subset of colluders of size at most c cannot produce the codeword of another user not in their group.

Definition 1. *([10]) A code Γ is called a c-frameproof code (c-FPC) if $Desc(C) \cap \Gamma = C$ for every subset $C \subseteq \Gamma$ of size at most c.*

A code Γ is called a secure frameproof code if two disjoint coalitions cannot produce the same descendant word.

Definition 2. *([10]) A code Γ is called a c-secure frameproof code (c-SFPC) if $Desc(C_1) \cap Desc(C_2) = \emptyset$ for any two disjoint subsets $C_1, C_2 \subseteq \Gamma$ of sizes at most c.*

Obviously, a c-SFPC is a c-frameproof code.

Definition 3. *A code Γ is called totally c-secure if there exists a tracing algorithm $A : Q'^{\ell} \to \Gamma$ such that $A(x) \in C$ for every $C \subseteq \Gamma$ of size at most c and $x \in F(C)$.*

It was proved in [3, 4] that totally c-secure codes do not exist when $c \geq 2$ and $n \geq 3$. A weakened form of totally secure codes is to allow the tracing to fail with a small chance.

Definition 4. *([3]) Let $\varepsilon > 0$. A code Γ is called c-secure with ε-error if there exists a tracing algorithm A satisfying condition: if $C \subseteq \Gamma$, $|C| \leq c$, creates a word $x \in F(C)$, then*

$$Pr[A(x) \in C] > 1 - \varepsilon. \tag{1}$$

Boneh et al [3, 4] gave a construction for c-secure codes which combines an inner c-secure code with an error-correcting code. The number of codewords of the inner code is much smaller than its length but in combination with the outer code results in a c-secure code whose length is logarithmic in the number of codewords. This is only an existence result and no explicit construction for the outer code with the required parameters has been given. A drawback of the inner code in Boneh et al's construction is that an innocent user may be accused and this will hold for the final construction as well. The chance of error can be made arbitrarily small but increasing the code length.

Other constructions [5, 9] of c-secure codes use the same structure but employ different inner codes. In [5] a family of 2-secure codes was proposed that uses the dual of the Hamming code as the inner code. The number of codewords of this inner code is nearly the same as its length and so the final code will have higher rate (ratio of the logarithm of the number of codewords to the length) compared to the construction in [3, 4]. Another advantage of this code is that the tracing algorithm uses the decoding algorithm of the dual of the Hamming code, and never outputs an innocent user. The number of codewords is 2^n and since the number of codewords of the inner code is the same as the alphabet size of the outer code, the higher rate is when the outer code is over $GF(2^n)$. In [9] a construction of a 3-secure codes using a class of inner codes called scattering codes, and an outer code which is a dual of the Hamming code is given. The tracing algorithm may output an innocent user and the code is shown to outperforms the code in [3, 4] for some parameters. This construction results in false accusation.

In all above constructions an 'inner code' is combined with an outer code which is an error-correcting code (dual Hamming code in that last construction). The inner code in the first construction is a 2-secure code with n codewords and length $(n-1)d$, and in the last one, is a scattering code with $2n$ codewords and

length $(2n+1)d$ and is the same as the first construction with an added first column. In Boneh et al construction, $d = 2n^2 \log(2n/\varepsilon)$. That is for n codewords, the length of the inner code is $\approx n^3(\log(2n/\varepsilon))$ and the code is n-secure.

In [4], with error ε, the inner code has n codeword and length $O(n^3 \log n/\varepsilon)$, the tracing algorithm may output innocent users. In [5], using dual binary Hamming code, a code of size n length n and error $2n/2^n$ is constructed. However, the code size must of the form $n = 2^i - 1$. In [9], a 3-secure code is introduced with code size of the same form $2^i - 1$. The length of this code is $(n-1)(2t+1)d$.

In the next section, we will construct a new 2-secure inner code with an arbitrary size n. Our tracing algorithm either fails or outputs a real colluder.

3 A New Inner Code

In this section we construct a binary code γ and prove that the code is a 2-SFPC. We give an efficient tracing algorithm and show that the code is a 2-secure code and calculate the error probability in tracing. We show that if the pirate word contains at least one mark '?' then the tracing algorithm correctly outputs a colluder.

The codewords are elements of the set $\{0,1\}^{n^2}$ and can be represented by $n \times n$ binary matrices. To construct the code, we choose n base-points b_1, b_2, $\ldots$, b_n, each point being a position of the $n \times n$ matrix such that there is exactly one base-point on each row and on each column. That is, if we assume the base-point b_i is on the row r_i and column c_i, then $(r_1, r_2, \ldots, r_n)$ and $(c_1, c_2, \ldots, c_n)$ are permutations of $(1, 2, \ldots, n)$. For a square matrix M of order n, we denote by $M(r, c)$ the entry in the r^{th} row and the c^{th} column. Now n codewords M_1, M_2, $\ldots$, M_n are constructed as follows.

For each i, $1 \leq i \leq n$, M_i is an $n \times n$ binary matrix whose (r, c) entry is given by

$$M_i(r,c) = \begin{cases} 1, \text{ if } r = r_i \text{ and } c \neq c_i \\ 1, \text{ if } r \neq r_i \text{ and } c = c_i \\ 0, \text{ otherwise} \end{cases}$$

For two base-points b_{i_1}, b_{i_2}, define

$$Rec(i_1, i_2) = \{(r, c) : r \in \{r_{i_1}, r_{i_2}\}, c \in \{c_{i_1}, c_{i_2}\}\}$$

$Rec(i_1, i_2)$ is the set of four vertices of the rectangle formed by the two base-points b_{i_1}, b_{i_2}. We call the pair of vertices (r_{i_1}, c_{i_2}) and (r_{i_2}, c_{i_1}) *opposite base-points* and denote by $Opp(i_1, i_2) = \{(r_{i_1}, c_{i_2}), (r_{i_2}, c_{i_1})\}$.

For any two codewords M_{i_1} and M_{i_2}, it is easy to see that the set of undetectable positions consists of four vertices of $Rec(i_1, i_2)$ together with all the positions that are not on rows r_{i_1}, r_{i_2} and not on columns c_{i_1}, c_{i_2}. The detectable positions are the positions on the rows r_{i_1}, r_{i_2} and columns c_{i_1}, c_{i_2}, except for the four positions of $Rec(i_1, i_2)$. The number of detectable positions is $4n - 8$.

$$D(C) = \{(r, c) : r \in \{r_{i_1}, r_{i_2}\} \text{ or } c \in \{c_{i_1}, c_{i_2}\}\} \setminus Rec(i_1, i_2)$$
$$U(C) = \{(r, c) : r \neq r_{i_1}, r_{i_2},\ c \neq c_{i_1}, c_{i_2}\} \cup Rec(i_1, i_2)$$

Theorem 1. *For a matrix $M \in \{0,1,?\}^{n^2}$, M is a member of $F(M_{i_1}, M_{i_2})$ if and only if*

1. *M has the values 0 on the two base-points b_{i_1} and b_{i_2}*

$$M(r_{i_1}, c_{i_1}) = M(r_{i_2}, c_{i_2}) = 0$$

2. *M has the values 1 on the two opposite base-points $Opp(i_1, i_2)$*

$$M(r_{i_1}, c_{i_2}) = M(r_{i_2}, c_{i_1}) = 1$$

3. *M has the values 0 on all the positions that are not on the row r_{i_1}, r_{i_2}, and not on the column c_{i_1}, c_{i_2}*

$$M(r, c) = 0 \quad \text{for all } r \neq r_{i_1}, r_{i_2}, c \neq c_{i_1}, c_{i_2}$$

Theorem 2. *For any $n \geq 4$, γ is a 2-secure frameproof code.*

Proof. Let M_{i_1}, M_{i_2}, M_{i_3} and M_{i_4} be four different codewords. ¿From Theorem 1, for any descendant M of $\{M_{i_1}, M_{i_2}\}$ and M' of $\{M_{i_3}, M_{i_4}\}$, $M(r_{i_1}, c_{i_2}) = M(r_{i_2}, c_{i_1}) = 1$ and $M'(r_{i_1}, c_{i_2}) = M'(r_{i_2}, c_{i_1}) = 0$. This implies $M \neq M'$.

Definition 5. *Let M be a binary matrix of size n. The set $ColluderPair(M)$ is defined as follows*

1. *a member of $ColluderPair(M)$ is a subset of $\{M_1, M_2, \ldots, M_n\}$ with two element*
2. *$\{M_{i_1}, M_{i_2}\} \in ColluderPair(M)$ if and only if*

(T1) $M(r_{i_1}, c_{i_1}) = M(r_{i_2}, c_{i_2}) = 0$,
(T2) $M(r_{i_1}, c_{i_2}) = M(r_{i_2}, c_{i_1}) = 1$, *and*
(T3) $M(r, c) = 0$, *for all* $r \neq r_{i_1}, r_{i_2}$, $c \neq c_{i_1}, c_{i_2}$.

$ColluderPair(M)$ is the set of all pairs of colluders that could have generated M

$$ColluderPair(M) = \{\{M_{i_1}, M_{i_2}\} : M \in Desc(M_{i_1}, M_{i_2})\}$$

3.1 Properties of $ColluderPair(M)$

In this section, we will look at the properties of the set $ColluderPair(M)$ which help us to derive tracing algorithm. We need the following lemma.

Lemma 2. *Suppose $\{S_1, S_2, \ldots, S_k\}$ is a collection of sets such that*

1. *Each set contains exactly two elements,*
2. *Any two sets have non-empty intersection, and*
3. *Union of all these sets contains more than three elements $|\cup_{i=1}^{k} S_i| > 3$*

then

$$\cap_{i=1}^{k} S_i \neq \emptyset$$

Proof. Assume $S_1 = \{x_1, x_2\}$, $S_2 = \{x_1, x_3\}$, $x_4 \in S_3$. Since S_3 has non-empty intersections with both S_1 and S_2, we must have $S_3 = \{x_1, x_4\}$. For any other set S_j, $4 \leq j \leq k$, since S_j has non-empty intersections with all three sets S_1, S_2 and S_3, S_j must contains x_1. Therefore,

Lemma 3. *Let M be given. If $S_1, S_2 \in ColludePair(M)$, then $S_1 \cap S_2 \neq \emptyset$.*

Proof. Follows from Theorem 2.

For any three base-points b_{i_1}, b_{i_2} and b_{i_3}, let $SM[i_1, i_2, i_3]$ be the binary matrix whose entries are all zeros except for the six opposite base-points $Opp(i_1, i_2)$, $Opp(i_2, i_3)$ and $Opp(i_3, i_1)$. ¿From now on, these matrices $SM[i_1, i_2, i_3]$ are called *special matrices*. It is easy to check that the special matrix $SM[i_1, i_2, i_3]$ belong to all three descendant sets $Desc(M_{i_1}, M_{i_2})$, $Desc(M_{i_2}, M_{i_3})$ and $Desc(M_{i_3}, M_{i_1})$. Special matrices are characterised by the following Lemma.

Lemma 4. $ColluderPair(M) = \{\{M_{i_1}, M_{i_2}\}, \{M_{i_2}, M_{i_3}\}, \{M_{i_3}, M_{i_1}\}\}$
if and only if $M = SM[i_1, i_2, i_3]$.

Proof. Firstly, if $\{M_{i_1}, M_{i_2}\}$, $\{M_{i_2}, M_{i_3}\}$, $\{M_{i_3}, M_{i_1}\} \in ColluderPair(M)$ then it follows from Definition 5 that the matrix M has all entries equal to zero except for the six opposite base-points $Opp(i_1, i_2)$, $Opp(i_2, i_3)$ and $Opp(i_3, i_1)$. That means $M = SM[i_1, i_2, i_3]$.
Conversely, if $M = SM[i_1, i_2, i_3]$ is a special matrix, then the only pairs that satisfy the three conditions (T1), (T2), (T3) are $\{i_1, i_2\}$, $\{i_2, i_3\}$ and $\{i_3, i_1\}$.

Theorem 3. *For any binary matrix M, if M is not a special matrix then*

$$\bigcap\{S : S \in ColluderPair(M)\} \neq \emptyset$$

And if $M = SM[i_1, i_2, i_3]$ then

$$ColluderPair(M) = \{\{M_{i_1}, M_{i_2}\}, \{M_{i_2}, M_{i_3}\}, \{M_{i_3}, M_{i_1}\}\}$$

Proof. Let $Collude(M) = \bigcup\{S : S \in ColluderPair(M)\}$ where M is a non-special matrix. Since each member of $ColluderPair(M)$ is a set that contains two elements and any two members of $ColluderPair(M)$ have non-empty intersection (Lemma 3), if $|Colluder(M)| > 3$ it follows from Lemma 2 that intersection of all members of $ColluderPair(M)$ is not empty.
If $|Colluder(M)| = 3$ then $ColluderPair(M)$ is either equal to $\{\{i_1, i_2\}, \{i_2, i_3\}\}$ or $\{\{i_1, i_2\}, \{i_2, i_3\}, \{i_3, i_1\}\}$. The latter cannot happen by Lemma 4 because M is not a special matrix. If $|Colluder(M)| = 2$, then $ColluderPair(M)$ has only one member. In all cases, we have $\bigcap\{S : S \in ColluderPair(M)\} \neq \emptyset$.

Since the pair of the actual colluders is included in the set $ColluderPair(M)$, if M is not a special matrix then from the above theorem the intersection of all members of $ColluderPair(M)$ is not empty. This intersection is a subset of the colluders.

3.2 Tracing Algorithm

Given a matrix M formed by two colluders. ¿From Theorem 3, we have the following trivial tracing algorithm. We consider two cases:
Case 1: M does not have a mark '?'

If M is a special matrix, $M = SM[i_1, i_2, i_3]$, then the two colluders are among i_1, i_2, i_3; in this case, the algorithm fails to identify them.

If M is not a special matrix, then we form the set $ColludePair(M)$ that contains all the pairs $\{i_1, i_2\}$ that satisfy (T1), (T2), (T3) in Definition 5.

A trivial method is to check all $\binom{n}{2}$ pairs $\{i_1, i_2\}$. In section 3.3, we use the properties of the set $ColludePair(M)$ to give a faster algorithm to search for such pairs. Theorem 3 ensures that the intersection of members of the set $ColludePair(M)$ is not empty. This intersection is the colluders. Output this intersection.
Case 2: M contains marks '?'

In this case, we always can find a colluder. Firstly, we substitute all the marks '?' by an arbitrary values 0 or 1 so that the resulting matrix M' is not a special matrix. One way to make this substitution easy is by observing that all special matrices have weight equal to 6. Therefore, when we substitute the marks '?' by 0 or 1, we need only to ensure that M' has weight not equal to 6 to guarantee that it is not a special matrix.

Since γ is a binary code, from Lemma 1, the binary matrix M' is a descendant matrix formed by the two colluders. As in case 1, form the set $ColludePair(M')$, and the colluder is in the intersection of all members of $ColludePair(M')$.
Tracing error: The only case when the tracing algorithm fails is when M is a special matrix.

Suppose that the two users 1 and 2 collude and they know that the tracing algorithm is deterministic if the pirate matrix contains at least a mark '?'. The number of special matrices that they can form is $n-2$. These matrices are $SM[1,2,3]$, $SM[1,2,4]$, ..., $SM[1,2,n]$. Since there are $4n-8$ detectable positions for $\{M_1, M_2\}$. ¿From *Embedding Assumption*, the best strategy that they have is replacing detectable positions with random marks correspond to 0 or 1. The total number of the matrices that they can form in this way is 2^{4n-8}. It follows that the tracing error is not larger than $\frac{n-2}{2^{4n-8}}$. However, if the colluders have no knowledge about the tracing algorithm then the tracing error is $\frac{n-2}{3^{4n-8}}$.

3.3 Faster Tracing

The main step in tracing algorithm is to determine the set $ColluderPair(M)$ for a non-special matrix M. The trivial solution requires at most $\binom{n}{2}$ steps. In this section, we present a faster tracing algorithm that use the weight of the matrices. The weight of a matrix is the number of ones in the matrix.

Theorem 4. *Let $M \in Desc(M_{i_1}, M_{i_2})$. If $weight(M) > 6$, then*

1. *there exist at least three 1's on some row or some column;*

2. *if a row or a column consists of at least three 1's then this row or column must contain one of the base-points* b_{i_1}, b_{i_2}.

Proof. The only places that we can find entries 1 in M are rows r_{i_1}, r_{i_2} or columns c_{i_1}, c_{i_2}. We know that at the two opposite base-points $(r_{i_1}, c_{i_2}), (r_{i_2}, c_{i_1})$, we have two 1's. Therefore, if there are at most two 1's on each row and column of M then we have at most four other 1's in these rows r_{i_1}, r_{i_2} and columns c_{i_1}, c_{i_2}. It follows that the weight of M cannot exceed 6.
Now suppose that there are at least three 1's in the same column. This column must be either column c_{i_1} or c_{i_2} as in the other columns there are at most two 1's. Similarly, if there are at least three 1's in the same row, this row must be either row r_{i_1} or r_{i_2}. This proves the second part of the theorem.

From the above theorem, we can see that if $weight(M) > 6$, we only need to identify a row or a column with three 1's. Since there is exactly one base-point in each row or column, the colluder's base-point will be identified.

If $weight(M) \leq 6$, then to determine the set $ColluderPair(M)$, using condition (T2), we only need to check for at most $\binom{6}{2} = 15$ pairs.

Theorem 5. *The tracing algorithm either outputs a correct colluder or fails with probability* $\frac{n-2}{2^{4n-8}}$. *If the pirate matrix contains at least one mark '?' then the algorithm correctly outputs a colluder.*

3.4 Reducing the Code Length

Since all codewords have the value 0 at base-points, we can remove n positions corresponding to the n base-points. Moreover, if we choose the n base-point to be $b_i = (i, i)$ then every codeword is a symmetric matrix. Therefore, we only need to record the lower part of the matrix and so the code has length $n(n-1)/2$.

4 Construction from Traceability Codes

In this section we combine the code $\gamma = \{M_1, M_2, \ldots, M_n\}$ constructed in the above section with 2-traceability codes to have 2-secure codes with shorter length. c-Traceability codes are defined as follows.

Definition 6. *([10]) Let* Γ *be an* n*-ary code of length* L, $C = \{u_1, \cdots, u_b\} \subseteq \Gamma$, *where* $u_i = (a_{i1}, a_{i2}, \cdots, a_{iL})$. Γ *is called* c*-traceability code, or* c*-TA code for short, if the following condition is satisfied: for any* $C \subseteq \Gamma, |C| \leq c$, *and any* $(x_1, \cdots, x_L) \in desc(C)$, *there is a* $u_i \in C$ *such that* $|\{j : x_j = a_{ij}\}| > |\{j : x_j = a_j\}|$ *for any* $(a_1, \cdots, a_L) \in \Gamma \setminus C$.

c-TA codes can tolerate some erased positions (positions with '?'). The bound on the maximum number of erasures tolerated by a c-TA code was given in [8]. Let Γ be an n-ary code and $C = \{u_1, \cdots, u_b\} \subseteq \Gamma, b \leq c$. Define a set

$$F(C; e) = \{(x_1, \cdots, x_L) \in F(C) : |\{j : x_j = ?\}| \leq e\}.$$

Theorem 6. *Let Γ be an $(L, N, D)_q$-ECC, and c be an integer.*

1. *([10])) If*
$$D > (1 - \frac{1}{c^2})L \tag{2}$$
then Γ is a c-TA code.
2. *([8]) If*
$$D > (1 - \frac{1}{c^2})L + \frac{e}{c^2} \tag{3}$$
then Γ is c-TA code tolerating e erasures.

Let Γ be an n-ary code of length L and size N over an alphabet $\{a_1, a_2, \ldots, a_n\}$. Define a binary code $\Delta(\Gamma, \gamma)$ in which each codeword has length ℓL, and obtained in the following way $U = M_{i_1} \| M_{i_2} \| \ldots \| M_{i_L}$, where $a_{i_1}\, a_{i_2}\, \ldots\, a_{i_L} \in \Gamma$.

Theorem 7. *Suppose γ is an (ℓ, n) c-secure code with ε-error, and Γ is an $(L, N, D)_n$ c-TA code satisfying (3). Then $\Delta(\Gamma, \gamma)$ is a c-secure code with error at most $(\varepsilon L)^{e+1}$.*

Proof. Denote by A_O, A_I the tracing algorithm for the outer code and the inner code. Define a tracing algorithm for code $\Delta(\Gamma, \gamma)$ as follows. Suppose a pirate word $X = X_1 \| X_2 \| \cdots \| X_L$ is given.
Step 1: Apply A_I to each X_j, $j = 1, 2, \cdots, L$. Suppose the output is M_{i_j}.
Step 2: Apply A_O to $a_{i_1} a_{i_2} \ldots a_{i_L}$. The output U of A_O is treated as a traitor. For this tracing, an error happens only if $|\{j : A_I(X_j) = \emptyset\}| > e$. While for A_I the tracing error is ε, so the tracing error for the code $\Delta(\Gamma, \gamma)$ is at most $\binom{L}{e+1}\varepsilon^{e+1} < (\varepsilon L)^{e+1}$.

The following is an examples of the resulting 2-secure codes.

Theorem 8. *Let n be a prime power, e, k be positive integers such that $k \leq \frac{1}{4}(n - e - 1)$. There exists a 2-secure code with error $\left((n-2)(n-1)/2^{4n-8}\right)^{e+1}$, the length of the code is $n^2(n-1)$ and the number of codewords is n^k.*

Proof. Let Γ be a Reed-Solomon code of length $L = n - 1$ and dimension k over $GF(n)$. Then from Theorem 7 $\Delta(\Gamma, \gamma)$ is a 2-secure code with error at most $\left((n-2)(n-1)/2^{4n-8}\right)^{e+1}$.

The following is a family of 2-TA codes. A code Γ is called *an equidistance code* if the Hamming distances between any two codewords are all the same.

Theorem 9. *Equidistant code with odd distance is 2-TA code.*

Proof. Let $X \in desc(U_1, U_2)$, then $d(X, U_1) + d(X, U_2) = d(U_1, U_2) = d$. Since d is odd, it follows that $d(X, U_1) \leq (d-1)/2$ or $d(X, U_2) \leq (d-1)/2$.

5 Construction from Perfect Hash Families

In this section we construct 2-secure codes with more codewords by combining a 2-secure code with a perfect hash family. Using this construction with the inner code given in Section 3 and a perfect hash family given in [1, 11] give a code with 7^{2^k} codewords and length 16×7^k.

Definition 7. *Let N, n, t be integers, X and Y be sets of size N and n, respectively, $\mathcal{F}$ be a family of s functions $f : X \to Y$. $\mathcal{F}$ is call a perfect hash family, denoted by PHF$(s; N, n, t)$, if for any subset $Z \subseteq X$ of size t, there exists an $f \in \mathcal{F}$ such that $f|_Z$ is one-to-one.*

Let γ be an (ℓ, n) code, $\mathcal{F} = \{f_1, f_2, \cdots, f_s : f_i : X \to Y\}$ be a PHF$(s; N, n, t)$. Define a code $\Omega(\gamma, \mathcal{F})$ consisting of N codewords of length $s\ell$. Each codeword in $\Omega(\gamma, \mathcal{F})$ is labelled by an element $x \in X$, and is defined by

$$u_{f_1(x)} \,\|\, u_{f_2(x)} \,\|\, \cdots \,\|\, u_{f_s(x)}$$

here $\|$ means concatenation, and $u_{f_j(x)} \in \Gamma$ for all j.

We consider a code $\Omega(\gamma, \mathcal{F})$, where $\gamma = \{M_1, M_2, \cdots, M_n\} \subseteq \{0,1\}^{n^2}$ is the code constructed in Section 3, $\mathcal{F} = \{f_1, f_2, \cdots, f_s : f_i : X \to Y\}$ is a PHF$(s; N, n, 4)$. Suppose $\mathcal{C} = \{U_1, U_2\} \subseteq \Omega(\gamma, \mathcal{F})$ is a collusion

$$U_i = M_{f_1(x_i)} \| M_{f_2(x_i)} \| \cdots \| M_{f_s(x_i)}, \quad i = 1, 2$$

Then the feasible set of $\mathcal{C}$ is given by,

$$F(\mathcal{C}) = \{X_1 \| X_2 \| \cdots \| X_s : X_j \in F(M_{f_j(x_1)}, M_{f_j(x_2)}), 1 \le j \le s\}$$

Every $X \in \{0,1\}^{sn^2}$ is naturally represented by $X = X_1 \| X_2 \| \cdots \| X_s$ with $X_j \in \{0,1\}^{n^2}$ for each j. For a given $X \in \{0,1\}^{sn^2}$, define

$$ColluderPair(X) = \{\mathcal{S} \subseteq \Omega(\gamma, \mathcal{F}) : |\mathcal{S}| = 2, X \in F(\mathcal{S})\}$$

Lemma 5. *Let $X \in \{0,1\}^{sn^2}$ be given. If $\mathcal{S}_1, \mathcal{S}_2 \in ColluderPair(X)$, then $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$.*

Proof. Assume $U_i = M_{f_1(x_i)} \| M_{f_2(x_i)} \| \cdots \| M_{f_s(x_i)}$ and $\mathcal{S}_1 = \{U_1, U_2\}$, $\mathcal{S}_2 = \{U_3, U_4\}$ are disjoint. Since $\mathcal{F}$ is a PHF$(s; N, n, 4)$, there exists an $f_j \in \mathcal{F}$ such that $f_j(x_1), f_j(x_2), f_j(x_3), f_j(x_4)$ are distinct and so the two sets $\{M_{f_j(x_1)}, M_{f_j(x_2)}\}$ and $\{M_{f_j(x_3)}, M_{f_j(x_4)}\}$ are disjoint. It follows that the two descendant sets of S_1 and S_2 are disjoint.

Lemma 6. *Let X be given. Then either*

$$\cap\{\mathcal{S} : \mathcal{S} \in ColluderPair(X)\} \neq \emptyset$$

or there exists distinct U_1, U_2, U_3 such that

$$ColluderPair(X) = \{\{U_1, U_2\}, \{U_2, U_3\}, \{U_3, U_1\}\}$$

Proof. Similar to the proof of Theorem 3.

Lemma 7. *If* $ColluderPair(X) = \{\{U_1, U_2\}, \{U_2, U_3\}, \{U_3, U_1\}\}$ *where* $X = X_1 \| X_2 \| \cdots \| X_s \in \{0,1\}^{sn^2}$ *and* $U_i = M_{f_1(x_i)} \| M_{f_2(x_i)} \| \cdots \| M_{f_s(x_i)}$, *then for each* $j = 1, 2, \ldots, s$, *either* X_j *is a special matrix or a codeword matrix.*

Proof. We have $X_j \in Desc(M_{f_j(x_1)}, M_{f_j(x_2)})$, $X_j \in Desc(M_{f_j(x_2)}, M_{f_j(x_3)})$ and $X_j \in Desc(M_{f_j(x_3)}, M_{f_j(x_1)})$ for each $j = 1, \ldots, s$.
If $M_{f_j(x_1)}$, $M_{f_j(x_2)}$ and $M_{f_j(x_2)}$ are three different codeword matrices then from Lemma 4, X_j must be a special matrix.
If $M_{f_j(x_1)}$, $M_{f_j(x_2)}$ and $M_{f_j(x_2)}$ are not distinct codewords, say $M_{f_j(x_1)} = M_{f_j(x_2)}$, then it follows from $X_j \in Desc(M_{f_j(x_1)}, M_{f_j(x_2)})$ that $X_j = M_{f_j(x_1)} = M_{f_j(x_2)}$. In this case, X_j is a codeword matrix.

Lemma 8. *The tracing error is* $\varepsilon = (\frac{2n-2}{2^{4n-8}})^s$.

Proof. For each $j = 1, \ldots, s$, the number of special matrix that two colluders can produce is $n - 2$. Therefore, there are $2n - 2$ possibilities that X_j is a special matrix or a codeword matrix. The probabilities of producing such a X_j is $\frac{2n-2}{2^{4n-8}}$. It follows that the probability to have $ColluderPair(X) = \{\{U_1, U_2\}, \{U_2, U_3\}, \{U_3, U_1\}\}$ is $(\frac{2n-2}{2^{4n-8}})^s$.

In the following we show the existence of the perfect hash family.

Theorem 10. *([1, 11]) There exists a* $PHF(7^{k+1}; 7^{2^k}, 4, 4)$ *for all* $k \geq 0$.

Theorem 11. *Let* k *be an integer. There exists a 2-secure code with error* $\varepsilon = \left(\frac{3}{2^7}\right)^{7^{k+1}}$ *of length* $L = 16 \times 7^{k+1}$ *and consisting of* $N = 7^{2^k}$ *codewords.*

Proof. Use $\mathcal{F}_k = PHF(7^{k+1}; 7^{2^k}, 4, 4)$ and the inner code γ with $n = 4$, $\Omega(\gamma, \mathcal{F}_k)$ is the 2-secure code with error $\left(\frac{2\times 4-2}{2^{4\times 4-8}}\right)^{7^{k+1}} = \left(\frac{3}{2^7}\right)^{7^{k+1}}$.

6 Comparison and Concluding Remarks

We considered 2-secure fingerprinting codes and presented a number of new constructions. A c-Secures code provides a tracing algorithm and an estimate of the highest probability of incorrect tracing. Our main construction, similar to all other known ones, have two layers. A 2-secure code is used as the inner code and then an outer structure is used to increase the number of codewords. All previous inner codes have shortcomings. Our proposed inner codes, improves on all the known codes by having a number of desirable properties simultaneously. Most importantly, it ensures that no other innocent users will be accused. The only other inner code that satisfy this property can exist for very limited range of number of codewords. Noting that this number is the alphabet size of the outer structure means that a much wider range of outer structures can be used and so better c-secure codes can be obtained. We show two general form of outer structures, one based on 2-TA codes and the second on perfect hash families and in both cases obtained the probability of incorrect tracing in terms of the parameters of the inner code and the outer structures.

References

[1] M. Atici, S. S. Magliveras, D. R. Stinson, and W. D. Wei. Some recursive constructions for perfect hash families. *Journal of Combinatorial Designs*, 4:353–363, 1996. 160, 161

[2] D. Boneh and M. Franklin. An efficient public key traitor tracing scheme. In *Advances in Cryptology - CRYPTO'99, Lecture Notes in Computer Science*, volume 1666, pages 338–353. Springer-Verlag, Berlin, Heidelberg, New York, 1999. 151

[3] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. In *Advances in Cryptology - CRYPTO'95, Lecture Notes in Computer Science*, volume 963, pages 453–465. Springer-Verlag, Berlin, Heidelberg, New York, 1995. 149, 150, 151, 152, 153

[4] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, Vol. 44, No. 5:1897–1905, 1998. 150, 151, 153, 154

[5] J. Domingo-Ferrer and J. Herrera-Joancomarti. Short collusion-secure fingerprints based on dual binary hamming codes. *Electronics Letters*, Vol. 36, No. 20:1697–1699, 2000. 153, 154

[6] A. Fiat and M. Naor. Broadcast encryption. In *Advances in Cryptology – CRYPTO'93, Lecture Notes in Computer Science*, volume 773, pages 480–491. Springer-Verlag, Berlin, Heidelberg, New York, 1994. 151

[7] A. Kiayias and M. Yung. Self protecting pirates and black-box traitor tracing. In *Advances in Cryptology - CRYPTO'01, Lecture Notes in Computer Science*, volume 2139, pages 63–79. Springer-Verlag, Berlin, Heidelberg, New York, 2001. 151

[8] R. Safavi-Naini and Y. Wang. Collusion secure q-ary fingerprinting for perceptual content. In *Security and Privacy in Digital Rights Management (SPDRM 2001), Lecture Notes in Computer Science*, volume 2320, pages 57–75. Springer-Verlag, Berlin, Heidelberg, New York, 2002. 158, 159

[9] F. Sebe and J. Domingo-Ferrer. Short 3-secure fingerprinting codes for copyright protection. In *Proceedings of ACISP'02, Lecture Notes in Computer Science*, volume 2384, pages 316–327. Springer-Verlag, Berlin, Heidelberg, New York, 2002. 150, 153, 154

[10] J. N. Staddon, D. R. Stinson, and R. Wei. Combinatorial properties of frameproof and traceability codes. *IEEE transactions on information theory*, Vol. 47, No. 3:1042–1049, 2001. 151, 152, 153, 158, 159

[11] D. Stinson, T. Trung, and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures. *Journal of Statistical Planning and Inference*, 86(2):595–617, 2000. 151, 160, 161

[12] D. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM Journal on Discrete Mathematics*, 11:41–53, 1998. 151

[13] D. R. Stinson and R. Wei. Key preassigned traceability schemes for broadcast encryption. In *Proceedings of SAC'98, Lecture Notes in Computer Science*, volume 1556, pages 144–156. Springer-Verlag, Berlin, Heidelberg, New York, 1999. 151